GADGET Documentation

Release 0.1

Peter Revelle

Apr 26, 2018

Contents

Gridded **A**tmospheric **D**ata Downscalin**G** and **E**vapotranspiration **T**ools for High-resolution Distributed Reference ET in Complex Terrain

Contents:

CHAPTER 1

Quickstart

CHAPTER 2

Notes

Gadget Note #0627 This day operates NM data

1. Beforehand, we got LinkeTurbidity offline. <- `Save_linke_turbidityNDimage`
2. Project LinkeTurbidity <- `GdalProjLinkeLatLong`
3. Down size them to NM scale or Walnute Gulch <-Save_linke_turbidity_NDimageMETDATA
4. Divide LinkeTurbidity by 20. (. for decimal floating)–> due to the algorithm, true value should around 2-3 ——————————-Record starts——————————————-
5. Script: `Save_linke_turbidity_NDimageMETDATA` Function: time_interpolation Interpolate cropped LinkeTurbidity Directly run the entire script Directory: 4 rows Linke data cropped according to DEM domain –>result in linke_METDATA_by_day Crop_Walnut: 5 rows Linke data cropped according to MeteData domain Ref_map: take the geo attribute of the map –> in first round, get one from the "crop" directory
6. Script: `GdalConvLinkeCropLatLong250` Crop the original LatLong World map to 250 m smaller area

#6.28 Really starts Walnut Gulch

1. Create a buffering layer for Walnut Gulch and failed.

#7.11 grass GIS –Creating Shading Area for mountains Dealing with Horizon and solar radiation

1. Grass LocationShadingArea
2. Terrain Analysis –> Horizon Angle [r.horizon]
3. Solar radiance and irradiation [r.sun] ———————————real start——————————- Solar radiation is based on horizon angles (Calculated later) and Linke Turbidity maps (Got before) Import LinkeTurbidity maps and use DEM to calculate horizon angles
4. Setting–>Region–>Display Region [g.region -p]
5. Change the working extents Same menu –> Set Region [g.region] –> type Bounds manually based on the values in spreadsheet "WalnutGulch_GrassExtents" equivalent to g.region n=32.95 s=31 e=-109.03 w=-111 res=0.00260605

6. File –> Import raster data –> common formats import [r.in.gdal] –>import the buffering dem mergeArc_30m_wgs84 in folder Walnut Gulch Raw Check the overwrite box if needed Then, go to Layers–> Zoom to layer –> can see the DEM Using the measing tools measure the distance: Note: the merged DEM was problematic that it doesn't quite have 50 km's buffer for each direction. But it should be fine.
7. To calculate horizon angle, directly type r.horizon in the console to call the GUI Name of input elevation raster map is the imported map Note: the way r.horizon calculate angle is starting from 90 degree and go counter-clockwise Because r.sun and this assume east is 0 For small Walnut Gulch, for better accuracy, doing every 5 degrees. For NM, it calculates every 10 degrees. -0.05 degree for raster boundary to get rid of the no data rim.

#7.18 Grass GIS Open GRASS GIS in new directory Walnut Gulch Set region t Walnut_250: 01_longlat_wgs84

1. File –> Import raster data –> common format import [r.in.gdal] –> Import longlat_wgs84 from crop_250_Walnut
2. Set region for GRASS to be the same as imported LinkeTurbidity map
3. Quit and recalculate in ShadingArea –> Small Area
4. Import the same map in Small Area –> set region
5. Import rasters from linke_250_by_day_Walnut ————take a while————————
6. raster –> terrain analysis –> slope and aspect –> Name for slope and aspect name: WalnutSlope/Aspect Note: The horizon angle we calculated before is the sun angle that below it, the area behind that elevation would be shaded.
7. GRASS_Script –> r_runTopog_daily_loopPMR Change the names for DEM, Aspect, Slope, horizonsteps = previously calculated degree interval from r.horizon Names: Rb - beam, Rd - diffuse, Rg - global, timestep - 0.25 (15min) –>0.1 (6min), procs = 3 means 3 core processors. Note: Sun moves 1 degree every 5 to 6 min Need to run this script based on r.sun package in GRASS GIS (install extension –> raster –> r.sun.daily)
8. Run the command: H:\GRASS_Scripts\r_sunTopog_daily_loopPMR.py The raw unit would be converted later now is ~kW/h or /day not sure
9. Output grass files to Tiff: Command: r.out.gdal balabala Usage see NM_rsunGdalDOYexact.py Line 30

#7.20 Continue GRASS GIS

1. GRASS GIS setting: ShadingArea, Mapset: SmallArea, Location name: Walnut Gulch (in region setting)
2. NM_rsunGdalDOYexact.py: Search: H:\GRASS\ShadingArea\SmallArea\cell –>where the original files are Path: H:\GRASS_scripts –> not actually been used. Scripts' location Path2 : H:\Walnut\rsun_250m–> The output_linke_write_NDimage from last time, final destination Note: beam radiation only depends on the sun and the terrain
3. Redo what we just did: recalculate those parameters with the resolution the same as meteologic data assuming flat surfaces.
4. New mapset: Walnut_lowRes
5. Metdata for entire Northern American: the file should in: H:\GADGET\DEM\Metdata_DEM_CONUS.tif but is not there now. Imported to GRASS —> But it is also not in GRASS –> Ask Peter for it when needed Note: the new DEM created here is correspond to meteorologic data grid PRISM - 800 m This product called Grid Metdata or just metdata ? - Merge meteorologist and prism dataset, get temperature data from PRISM sampled at 4 km.
6. Starting with import the Northern US DEM to GRASS, manually specify the extent the same as Rb: -110.1900273269444455, 31.6392869372222201, -109.5463329769444414, 31.811286272222231
7. Import LinkeTurbidity data: \GADGET\LInkeTurbidity\linke_METDATA_by_day(This is NM one)

8. Open Script in that Folder: Save_linke_turbidity_NDimageMETDATA Walnut folder: linke_METDATA_by_day_Walnut Rows and Columns need to be changed: Latitude_index = 139 (NM) 5 (Walnut) longitude_index = 154 (NM) 16 (Walnut)
9. Repeat steps in the first day for NM applying to Walnut Gulch: Script:GdalConvLinkeCropLatLong250.py Des: crop_Walnut Change gdalwarp: 0.041666 –> 4k The size doesn't quite match –> make the domain bigger –> add command: -cutline %s -crop_to_cutline –> doesn't work –> tried lots of different ways –> the problem doesn't solve that day –> Peter took care of making the full coverage lowRes DEM

#7.25

1. Already got the right DEM –> mergeDEM_4km@Walnut_lowRes
2. Calculate the flat r.sun things same as before, need coarse grid slope and aspect: DEM4k_slope DEM4k_aspect trick the system by using the raster calculator: mergeDEM_4km@Walnut_lowRes *0 The rsunflat script is the newest version –> don't worry –> The script is basically the same - just changed the inputs
3. To import linke turbidity Crop_Walnut: coarse linke turbidity –> correspond to WalnutGulchRaw\boundary\entire_polygon.shp But this domain is not quite right –> Right one: mergeDEM_4km Use the entire_polygon to crop all linke data that attached to the line, the final result should be the same as the right domain –> script GdalConvLinkeCropLatLong.py has the right setting Now, the coarse resolution linke data are in: H:\GADGET\LinkeTurbidity\linke_METDATA_by_day_Walnut
4. Interpolate the new linke mapset by the same script as before –> don't forget to change your directory –> Now, import these to GRASS
5. GdalCOnvFlatRadEm2 –> final step to convert the rasters into the right unit
6. While the thing is running –> download METDATA from cida.usgs.com from 2000 to 2015 Line 47 of the script (download_METDATA_2014_15.py) after the ? are all the data we download Remember to change the domain!!!!
7. Now, the GRASS thing is done –> output_linke_write_NDimage those Rbflats by using the script NM_rsunflat_GdalDOYexact.py into \Walnut\rsun_METDATA
8. Corrections of slope and aspect settings –> already mentioned in 2. so no worries
9. Convert outputs to correct unit by using GdalConvFlatRadWm2.py into folder: FlatRadWm2
10. However, this time the output_linke_write_NDimage has issue –>probably influenced by the linke turbidity and aspect map.

#8.01 Continue fixing the wrong output_linke_write_NDimage

1. Use raster calculator to calculate DEM_sealevel_4km from our domain
2. Script to operate this is in sealevel folder keep using the same DEM as last time
3. No OK –> specify linke_value in grass run command to be 3.0 –>No–>several trials –> problem is on slope and aspect
4. Just use the flat DEM for slope and aspect –> works! –> just need to use pure blank
5. Everything is the same as before

Finished all preprocessing, and run the gadget! #8.03

1. download meteological data based on information mentioned on #7.25 Files stored in: H:\Walnut\METDATA Gadget assumes each file has a month of data The data could be viewed in GIS –> more like having bands storing different area. It is based on a combination of NLDAS and PRISM dataset.
2. Newest version of GADGET: REFET from Git
3. First, open: gadget_METDATAevap.ini to process downloaded data

4. Move Flat and Slope run in W/m2 into folderrsunWm2
5. Gadget lib: heart and core to gadget algorithm
6. To match the main GADGET functionality, create a new folder and resample the DEM: Take the first day DEM processed by ArcGIS Use the Script in GADGET folder: GdapConvDEMCropLatLong250.py Input folder: DEM output_linke_write_NDimage folder: dem_clip Don't forget to get your 366th day Finally flatdir = 'FlatRad_DEMRes'
7. Run it from terminal "The METDATA script": Look the comment for syntax for run: Python gadget_METDATAvap.py -I gadget_METDATAevap.ini -S 0 -E In this comment the -S and -E are referring back to the ini date
8. All output_linke_write_NDimage in folder: PM_RAD –> PM is the ref ET, rad –> radiation
9. These outputs are still in latlong: new boundary would delete two grids for each of the side

CHAPTER 3

Indices and tables

- genindex
- modindex
- search